

**IN THE SPECIFICATION:**

Please replace the paragraph beginning at line 17, p. 12 with the following:

β1  
It is understood that there are other ways to initiate the in thread debugger. For example, the in-thread debugger 152 may be inserted in into the process space of a running program and given the thread context of the running program, provided that the user has the proper access permissions. Referring to the flowchart of FIG. 4, and the diagrams of FIGS. 5-7, an example of how the in-thread debugger may be executed within a thread's context according to the teachings of the invention will be described, using the environment of FIG. 2 and the thread 130 of FIG. 3 as a starting point.

Please replace the paragraph beginning at line 3, p. 13 with the following:

BZ  
Turning to FIG. 4, the thread 130 (FIG. 5) is halted due to a previously defined breakpoint or due to an error at step 500. At step 502, a conventional debugger 158 allocates a block 155 of memory inside the process 122. In the WINDOWS 2000 operating system environment, this may be accomplished through the use of the "VirtualAllocEx" Application Programming Interface (API). At step 504, the conventional debugger 158 creates a loader module 157 for loading an in-thread debugger into the process 122. In creating the loader module 157 the conventional debugger 158 reads the current value of the instruction pointer 142 and writes the value into the loader module 157. The loader module 157 may also include: the filename file name of an in-thread debugger module, the name of the in-thread debugger entry point, the name and port of the machine with which the in-thread debugger will communicate during the debugging session, and the code for loading the in-thread debugger.